

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for determining if an object in a memory resource in a computer system is a candidate for garbage collection, comprising:

for each of a plurality of objects in the memory resource, maintaining a reference count based on a number of objects pointing thereto, and maintaining a depth value based on a distance from a global data object in a reference chain, the step of maintaining a depth value further including at least keeping an existing depth value associated with an object when a new link is added from another object to the object if the existing depth value associated with the object is a valid depth value and if the existing depth value associated with the object is an uninitialized value, changing the existing depth value associated with the object to one more than a depth value associated with said another object; and

identifying, based on the associated reference count and depth value, which of the plurality of objects are processed to determine whether or not they are garbage;

wherein each of the plurality of objects can be pointed to by one or more other objects and wherein each of the plurality of objects can be part of, and have different positions in, different reference chains such that the reference count and depth value change as the other objects that point to each of the plurality of objects change.

2. (Original) The method of claim 1, wherein:

for each object, the depth value is maintained by providing a depth field in metadata associated with the object.

3. (Currently Amended) The method of claim 1, wherein:

a first of the objects is identified for processing when it is determined that a second of the objects previously pointing thereto is no longer pointing thereto, and the depth value of the second object is one less than the depth value of the first object and the first of the objects is identified as not a garbage collection candidate even when it is determined that the second of the objects previously pointing thereto is no longer pointing thereto and the depth value of the second object is not one less than the depth value of the first object.

4. (Previously Presented) The method of claim 3, wherein:

the second object is no longer pointing to the first object because the second object was destroyed.

5. (Previously Presented) The method of claim 3, wherein:

the second object is no longer pointing to the first object because the second object was reset to point to another object.

6. (Original) The method of claim 1, wherein:

the distance for each of the plurality of objects is based on a number of objects linking the each of the plurality of objects to the global data object.

7. (Original) The method of claim 6, wherein:

the objects processed to determine whether or not they are garbage are processed by a loop detection mechanism for a reference counting garbage collector.

8. (Original) The method of claim 1, further comprising:

initializing the depth value of a new object created at runtime to one more than the depth value of an object that points to the new object and links the new object to the global data object.

9. (Original) The method of claim 1, wherein

when a particular one of the objects that is processed is determined to be garbage, its associated portion of the memory resource is made available for re-allocation.

10. (Original) The method of claim 1, wherein

when a particular one of the objects that is processed is determined to be non-garbage, its depth value, and the depth values of other ones of the objects that are pointed to by the particular object and linked to the global data object thereby, are reset.

11. (Original) The method of claim 10, wherein:

the depth values of objects that are pointed to by the other ones of the objects whose depth values are reset, and linked to the global data object thereby, are also reset.

12. (Original) The method of claim 10, wherein:

the particular object is determined to be non-garbage when it is accessible to outside objects.

13. (Original) The method of claim 1, wherein:

when a new link is created from a first object of the plurality of objects to a second object of the plurality of objects, and the first object has a valid depth value but the second object does not have a valid depth value, the depth value of the second object is initialized to one more than the depth value of the first object.

14. (Previously Presented) The method of claim 1, further comprising:
initializing a depth value of the global object to a valid depth value.

15. (Original) The method of claim 1, further comprising:
initializing depth values of the plurality of objects to a specified value.

16. (Currently Amended) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps of determining if an object in memory is garbage, the method steps comprising:

for each of a plurality of objects in the memory resource, maintaining a reference count based on a number of objects pointing thereto, and maintaining a depth value based on a distance from a global data object in a reference chain, the step of maintaining a depth value further including at least keeping an existing depth value associated with an object when a new link is added from another object to the object if the existing depth value associated with the object is a valid depth value and if the existing depth value associated with the object is an uninitialized value.

changing the existing depth value associated with the object to one more than a depth value associated with said another object; and

identifying, based on the associated reference count and depth value, which of the plurality of objects are processed to determine whether or not they are garbage;

wherein each of the plurality of objects can be pointed to by one or more other objects and wherein each of the plurality of objects can be part of, and have different positions in, different reference chains such that the reference count and depth value change as the other objects that point to each of the plurality of objects change.

17. (Currently Amended) A computer system for determining if an object in memory is garbage, comprising:

a processor and memory for maintaining, for each of a plurality of objects in the memory resource, a reference count based on a number of objects pointing thereto, and maintaining a depth value based on a distance from a global data object in a reference chain, said maintaining a depth value further including at least keeping an existing depth value associated with an object when a new link is added from another object to the object if the existing depth value associated with the object is a valid depth value and if the existing depth value associated with the object is an uninitialized value, changing the existing depth value associated with the object to one more than a depth value associated with said another object, and for identifying, based on the associated reference count and depth value, which of the plurality of objects are processed to determine whether or not they are garbage;

wherein each of the plurality of objects can be pointed to by one or more other objects and wherein each of the plurality of objects can be part of, and have different positions in, different

reference chains such that the reference count and depth value change as the other objects that point to each of the plurality of objects change.

18. (Previously presented) The method claim 1, wherein the depth value ranges from a first value to a predetermined maximum value and when a depth value associated with a first object reaches the predetermined maximum value, a depth value associated with a second object that links from the first object is wrapped around to the first value.
19. (Previously presented) The program storage device of claim 16, wherein the depth value ranges from a first value to a predetermined maximum value and when a depth value associated with a first object reaches the predetermined maximum value, a depth value associated with a second object that links from the first object is wrapped around to the first value.
20. (Previously presented) The computer system of claim 17, wherein the depth value ranges from a first value to a predetermined maximum value and when a depth value associated with a first object reaches the predetermined maximum value, a depth value associated with a second object that links from the first object is wrapped around to the first value.
21. (Previously presented) The method of claim 1, wherein a first of the objects is identified for processing when it is determined that a second of the objects previously pointing thereto is no longer pointing thereto, and the depth value of the first object is not known.
22. (Previously presented) The method of claim 1, further including:

identifying a first of the objects as not being garbage when it is determined that a second of the objects previously pointing thereto is no longer pointing thereto, if the depth value of the first object is known and the depth value of the second object is not one less than the depth value of the first object.